

Implementation of the Simple Query System with PHP

COMP 5120/6120

Haiquan Chen

Outline

- PHP Fundamentals
- Implementation of Your Query System

Outline

- PHP Fundamentals
- Implementation of Your Query System

What is PHP?

- PHP is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

```
<? echo "HI!"; ?>
```

Why PHP?

- Rationale
 - HTML layout language, without variables
 - Limitation of HTML, static, **no customizability**.
 - PHP, scripting, dynamic webpage,
 - Different webpage for different user inputs (adaptability)
 - Different webpage for different system usages (adaptivity)
 - Amazon, ...

How it works

- PHP is installed on web server
- Web server can be Apache
- Server parses files based on extensions
- Returns plain HTML, no code.

How To – The Basics

- Need to name files as a .php extension
Example: index.php, mypage.php
- Open and close tags: `<? ?>`
Was: `<?php ?>`
- Save file to server, view in a browser

Variables

In PHP, you create a variable with a dollar sign and some text.

Usually the text will be something descriptive of what it is going to hold.

```
$name = "John Smith";
```

```
$dept = "CS";
```

Variables

There are many different kinds of variables in PHP

- Scalar
- Array
- Object

Scalar Variables

Hold single values

- String/text
- Numbers

```
$name = "John";  
$dob = "1/1/2007";  
$age = 84;
```

Array Variables

Hold multiple values

All in one step example:

```
$kids = Array("Tom","Dick","Harry");
```

Multiple steps example:

```
$kids = Array();
```

```
$kids[0] = "Tom";
```

```
$kids[1] = "Dick";
```

```
$kids[2] = "Harry";
```

Individual array values are just a scalar

Hello World

helloworld.php

```
<html>
```

```
<body>
```

```
<? echo "Hello World!"; ?>
```

```
</body>
```

```
</html>
```

An Example to Randomize Slogan

- `<html>`
- `<body>`
- `<center>TEST

`
- `</center>`
- `<?php`
- `$value=rand(1,7);`
- `switch ($value) {`
- `case 1: print "I am going to get an A in Databases. \n
";`
- `break ;`
- `case 2: print "To be or not to be, that is the question ... \n
";`
- `break ;`

An Example to Randomize Slogan

- `case 3: print "Today is a great day! \n
";`
- `break ;`
- `case 4: print "We are what we repeatedly do.\n
";`
- `break ;`
- `case 5: print "Databases are the best! \n
";`
- `break ;`
- `case 6: print "Auburn University is number one. \n
";`
- `break ;`
- `case 7: print "I love football! \n
";`
- `break ;`
- `}`
- `?>`
- `</body>`
- `</html`
- `>`

<http://www.auburn.edu/~chenhai/test/test.php>

Outline

- PHP Fundamentals
- Implementation of Your Query System

Your Query System

So far, PHP fundamentals, then how to use them to **implement** your query system?

- consists of
 - a HTML file to request a query
 - a PHP file to execute the query (connecting to database first) and display the result
- A preferred way: interleave them in a single file**

Query System Decomposition

- A HTML send-query interface:
 - data_sender.html

```
<form method="POST" action="data_receiver.php">  
<input type="text" name="name">  
<input type="text" name="age">  
<input type="submit">  
</form>
```

Query System Decomposition

- A PHP receive-query-and-display-result interface
 - data_receiver.php data captured

Variables: like containers

- \$_POST['name']
 - Automatically assigned with the value of <input type="text" name="name">
- \$_POST['age']
 - Automatically assigned with the value of <input type="text" name="age">

HTML as the output of PHP

receiver.php

```
<html><body>
<?
    print "My name is $_POST['name'] and I am $_POST['age'] years old";
?>
</body></html>
```

Connecting to DB from PHP

Create one connection script:

dbconn.php

```
<?
    mysql_connect($server,$user,$pw);
    mysql_select_db($db);
?>
```

Connecting to DB from PHP

At the top of each file that needs the DB:

```
<? require("dbconn.php"); ?>
```

Dealing with INSERT

```
<?
    mysql_connect($server,$user,$pw);
    mysql_select_db($db);

    $name = $_POST['name'];
    $age = $_POST['age'];
    $sql = "INSERT into info (name,age) values('$name',
    '$age');";
    $success = mysql_query($sql);
?>
<html><body>
```

Dealing with INSERT

```
<? if($success)
{ echo "Thank you, your name and age were received."; }
else
{ echo "Sorry, your info wasn't received."; }
?>
</body></html>
```

Dealing with SELECT

```
<?
mysql_connect($server,$user,$pw);
mysql_select_db($db);

$sql = "SELECT * FROM info";
$result = mysql_query($sql);
?>
```

Dealing with SELECT

```
<html><body>
<table border="1">
<? while($array = mysql_fetch_array($result))
  {   echo "<tr><td>";
      echo $array[0];
      echo "</td><td>";
      echo $array[1];
      echo "</td></tr>";
  }
?>
<table>
</body></html>
```

Some Hints

- Pay attention to the delivered arguments
 - System automatically replaces spaces with slashes (\$sql)
 - Use some function to strip slashes
- Identify the type of the passed SQL statement
 - Create, Insert, Update or Select ?
- Special functions
 - to get the attribute names
 - to get the count of rows

References

- www.brown.edu/webmaster/webpublishing/IntroToPHP.ppt
- <http://devzone.zend.com/manual>